

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/819,566	03/27/2001	James R. Van Eaton	143768.176.1	2623

7590 07/15/2004

RICK D. NYDEGGER  
WORKMAN, NYDEGGER & SEELEY  
1000 Eagle Gate Tower  
60 East South Temple  
Salt Lake City, UT 84111

EXAMINER

BLACKWELL, JAMES H

ART UNIT

PAPER NUMBER

2176

DATE MAILED: 07/15/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

## Office Action Summary

Application No.

09/819,566

Applicant(s)

VAN EATON ET AL.

Examiner

James H Blackwell

Art Unit

2176

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 27 March 2001.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-35 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-35 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 27 March 2001 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
Paper No(s)/Mail Date 2.03/27/01.
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_.
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: \_\_\_\_\_.

## DETAILED ACTION

### *Claim Rejections - 35 USC § 103*

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 1-33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Brooke et al (hereinafter Brooke, U.S. Patent No. 6,748,569).

In regard to independent Claim 1 (and similarly independent Claims 29, and 32), Brooke teaches XSP scripts, such as XSP script (300), that are used to generate classes of web pages. In one embodiment, XSP script (300) is initiated in XSP processor (202) by a user or another program requesting the URL of the script, such as the URL `http://server/xsp/inventors.xsp?l=en&target=ie5`, where, in this example, the source for the script is stored in a file named `inventors.xsp` in the `xsp` directory on computer system server. The request also includes parameters that provide information pertaining to, among other things, the XML and XSL documents that will be generated during data gathering process (204) and style sheet builder process (206) shown in Fig. 2. For example, the parameter `l=en` (language=english) provides information on which language will be used to display the resulting page. Another example is the parameter `target=ie5` that identifies the browser that will be used so that the documents can be tailored to the unique features and extensions of the particular browser. Other parameters may be implemented as required (Col. 7, lines 53-67; Col. 8, lines 1-3;

compare with Claim 1 (and similarly Claims 29, and 32), ***“in a system including a data server and a client that is associated with a web browser, wherein the client accesses data stored on the data server and displays such data using the web browser, a method for transforming the accessed data into a format for viewing using the web browser, the method comprising: an act of the client accessing a view descriptor, the view descriptor identifying stored data and including formatting parameters on how the identified stored data should be arranged when viewed; an act of the client processing the view descriptor using a generic view class to generate a specific view class; an act of the client accessing the stored data that was identified by the view descriptor; and an act of the client formatting the accessed data for viewing in accordance with the formatting parameters”***). It is noted that the majority of actions taught by Brooke are performed on a server rather than a client. However, it would have been obvious to one of ordinary skill in the art at the time of invention to perform such actions on the client because a client can also host a server and perform all the actions as claimed, on a single machine. A benefit would have been to use such a system for testing prior to deployment.

In regard to dependent Claim 2, Brooke teaches that the request also includes parameters that provide information pertaining to, among other things, the XML and XSL documents that will be generated during data gathering process (204) and style sheet builder process (206) shown in Fig. 2. For example, the parameter `l=en` (language=english) provides information on which language will be used to display the resulting page. Another example is the parameter `target=ie5` that identifies the browser

that will be used so that the documents can be tailored to the unique features and extensions of the particular browser (Col. 7, lines 60-67; Col. 8, lines 1-2; compare with Claim 2, “... ***an act of the client accessing a view descriptor that includes parameters on how the identified stored data should be arranged when viewed on the web browser***”). It is noted that the majority of actions taught by Brooke are performed on a server rather than a client. However, it would have been obvious to one of ordinary skill in the art at the time of invention to perform such actions on the client because a client can also host a server and perform all the actions as claimed, on a single machine. A benefit would have been to use such a system for testing prior to deployment.

In regard to dependent Claim 3, Brooke teaches that an XSP script is a document written in XSP and XML that specifies one or more XML documents as input to XSP processor (202). The output of server pages processing apparatus (200) is an XML object that may be a well-formed markup language document that is ready to be input to a browser, or it may be expressed in another format that is ready for use in a data-interchange process with another system, or for formatting and delivery to an HTML or non-HTML client (Col. 8, lines 7-15; compare with Claim 3, “... ***an act of the client accessing a view descriptor that includes Extensible Markup Language tags***”). It is noted that the majority of actions taught by Brooke are performed on a server rather than a client. However, it would have been obvious to one of ordinary skill in the art at the time of invention to perform such actions on the client because a client

can also host a server and perform all the actions as claimed, on a single machine. A benefit would have been to use such a system for testing prior to deployment.

In regard to dependent Claim 4, Brooke teaches XSP script (300) is initiated in XSP processor (202) by a user or another program requesting the URL of the script, such as the URL `http://server/xsp/inventors.xsp?target=ie5`, where, in this example, the source for the script is stored in a file named `inventors.xsp` in the `xsp` directory on computer system server (Col. 7, lines 54-60; compare with Claim 4, “... ***an act of the client downloading the view descriptor***”). It is noted that the majority of actions taught by Brooke are performed on a server rather than a client. However, it would have been obvious to one of ordinary skill in the art at the time of invention to perform such actions on the client because a client can also host a server and perform all the actions as claimed, on a single machine. A benefit would have been to use such a system for testing prior to deployment.

In regard to dependent Claims 5-7, Brooke teaches that an example of a typical Internet connection found in the prior art is shown in Fig. 1. A user that wishes to access information on the Internet typically has a computer workstation (112), also referred to as “the client workstation” that executes an application program known as a web browser (114). Workstation (112) establishes a communication link (116) with web server (118) such as a dial-up wired connection with a modem, a direct link such as a T1 or ISDN line, a wireless connection through a cellular or satellite network, or a local data transport system such as Ethernet or token ring over a local area network (Col. 4, lines 6-16; compare with Claim 5, “... ***an act of the client downloading the view***”).

***descriptor over a dial-up connection***” and Claim 6, “... ***an act of the client downloading the view descriptor over a permanent network connection***” and Claim 7, “... ***an act of the client downloading the view descriptor over the internet***”). It is noted that the majority of actions taught by Brooke are performed on a server rather than a client. However, it would have been obvious to one of ordinary skill in the art at the time of invention to perform such actions on the client because a client can also host a server and perform all the actions as claimed, on a single machine. A benefit would have been to use such a system for testing prior to deployment.

In regard to dependent Claim 8, does not teach *an act of the client accessing a locally stored view descriptor through an API to an operating system running on the client*. However, Brooke does teach that Fig. 2 shows a block diagram of server pages processing apparatus (200) for generating and processing XML and XSL documents using XML server pages language (XSP) in accordance with the present invention. The present invention includes the XSP script language and XSP processor (202) that interprets scripts written in XSP. XSP processor (202) resides on web server (116) (Fig. 1), and on other web servers, as required. One way to invoke XSP processor (202) is dynamically, through an online mechanization wherein a user submits one or more XSP scripts from client workstation (112) (Fig. 1) to XSP processor (202) through an application program interface (API) on web server (116) (Col. 7, lines 20-31). It is noted that the majority of actions taught by Brooke are performed on a server rather than a client. However, it would have been obvious to one of ordinary skill in the art at the time of invention to perform such actions on the client because a client can also host a



server and perform all the actions as claimed, on a single machine. A benefit would have been to use such a system for testing prior to deployment.

In regard to dependent Claim 9, Brooke teaches XSP scripts, such as XSP script (300), that are used to generate classes of web pages. In one embodiment, XSP script (300) is initiated in XSP processor (202) by a user or another program requesting the URL of the script, such as the URL `http://server/xsp/inventors.xsp?target=ie5`, where, in this example, the source for the script is stored in a file named `inventors.xsp` in the `xsp` directory on computer system server. The request also includes parameters that provide information pertaining to, among other things, the XML and XSL documents that will be generated during data gathering process (204) and style sheet builder process (206) shown in Fig. 2 (Col. 7, lines 53-64; compare with Claim 9, “... ***an act of the client processing the view descriptor and a generic view class associated with the view descriptor in order to generate a specific view class for the stored data***”). It is noted that the majority of actions taught by Brooke are performed on a server rather than a client. However, it would have been obvious to one of ordinary skill in the art at the time of invention to perform such actions on the client because a client can also host a server and perform all the actions as claimed, on a single machine. A benefit would have been to use such a system for testing prior to deployment.

In regard to dependent Claims 10 and 11, Brooke teaches that an XSP script is a document written in XSP and XML that specifies one or more XML documents as input to XSP processor (202). The output of server pages processing apparatus (200) is an XML object that may be a well-formed markup language document that is ready to be

input to a browser, or it may be expressed in another format that is ready for use in a data-interchange process with another system, or for formatting and delivery to an HTML or non-HTML client (Col. 8, lines 7-15). In addition, Brooke teaches XSP scripts, such as XSP script (300), that are used to generate classes of web pages. In one embodiment, XSP script (300) is initiated in XSP processor (202) by a user or another program requesting the URL of the script, such as the URL `http://server/xsp/inventors.xsp?target=ie5`, where, in this example, the source for the script is stored in a file named `inventors.xsp` in the `xsp` directory on computer system server. The request also includes parameters that provide information pertaining to, among other things, the XML and XSL documents that will be generated during data gathering process (204) and style sheet builder process (206) shown in Fig. 2 (Col. 7, lines 53-64; compare with Claim 10, “... ***an act of the client processing the view descriptor and a generic view class associated with the view descriptor, the view descriptor, including Extensible Style Language tags, in order to generate a specific view class for the stored data***” and Claim 11, “... ***an act of the client generating a specific view class***”). It is noted that the majority of actions taught by Brooke are performed on a server rather than a client. However, it would have been obvious to one of ordinary skill in the art at the time of invention to perform such actions on the client because a client can also host a server and perform all the actions as claimed, on a single machine. A benefit would have been to use such a system for testing prior to deployment.

In regard to dependent Claims 12-15, Brooke teaches that line (308) in XSP script (300) includes XSP control statement <xsp:query> to execute an XQL query, such as querying data island xsp.xml shown in Fig. 5. The <xsp:query> statement can be used to obtain a piece of XML from another document or an embedded data island. The source data can be queried to reduce the scope of the data returned. It can also be transformed using XSL patterns, which are either passed inline within the <xsp:query> tags, or by reference to an XSL document in the transform attribute (Col. 10, lines 7-14; compare with Claim 12, ***"... an act of the client processing the view descriptor to generate a query, which may be submitted to the data server"*** and Claim 13, ***"... an act of the view control processing the view descriptor to generate an SQL query, which may be submitted to the data server"*** and Claim 14, ***"... an act of the client submitting a query to the data server"*** and Claim 15, ***"... an act of the client submitting a SQL query to the data server"***). It is noted that the majority of actions taught by Brooke are performed on a server rather than a client. However, it would have been obvious to one of ordinary skill in the art at the time of invention to perform such actions on the client because a client can also host a server and perform all the actions as claimed, on a single machine. A benefit would have been to use such a system for testing prior to deployment.

In regard to dependent Claim 16, Brooke teaches XSP scripts, such as XSP script (300), that are used to generate classes of web pages. In one embodiment, XSP script (300) is initiated in XSP processor (202) by a user or another program requesting the URL of the script, such as the URL `http://server/xsp/inventors.xsp?target=ie5,`

where, in this example, the source for the script is stored in a file named inventors.xsp in the xsp directory on computer system server. The request also includes parameters that provide information pertaining to, among other things, the XML and XSL documents that will be generated during data gathering process (204) and style sheet builder process (206) shown in Fig. 2 (Col. 7, lines 53-64; compare with Claim 16, “... ***an act of the client downloading the stored data from the data server***”). It is noted that the majority of actions taught by Brooke are performed on a server rather than a client. However, it would have been obvious to one of ordinary skill in the art at the time of invention to perform such actions on the client because a client can also host a server and perform all the actions as claimed, on a single machine. A benefit would have been to use such a system for testing prior to deployment.

In regard to dependent Claims 17 and 18, Brooke teaches that an XSP script is a document written in XSP and XML that specifies one or more XML documents as input to XSP processor (202). The output of server pages processing apparatus (200) is an XML object that may be a well-formed markup language document that is ready to be input to a browser, or it may be expressed in another format that is ready for use in a data-interchange process with another system, or for formatting and delivery to an HTML or non-HTML client (Col. 8, lines 7-14; compare with Claim 17, “... ***an act of the client receiving stored data which includes Extensible Markup Language tags, from the server***” and Claim 18, “... ***an act of the client receiving stored data, which includes Hypertext Markup Language tags, from the server***”). It is noted that the majority of actions taught by Brooke are performed on a server rather than a client.

However, it would have been obvious to one of ordinary skill in the art at the time of invention to perform such actions on the client because a client can also host a server and perform all the actions as claimed, on a single machine. A benefit would have been to use such a system for testing prior to deployment.

In regard to dependent Claim 19, Brooke does not specifically teach, *an act of the client accessing locally stored data through an API to an operating system running on the client*. However, Brooke does teach that Fig. 2 shows a block diagram of server pages processing apparatus (200) for generating and processing XML and XSL documents using XML server pages language (XSP) in accordance with the present invention. The present invention includes the XSP script language and XSP processor (202) that interprets scripts written in XSP. XSP processor (202) resides on web server (116) (Fig. 1), and on other web servers, as required. One way to invoke XSP processor (202) is dynamically, through an online mechanization wherein a user submits one or more XSP scripts from client workstation (112) (Fig. 1) to XSP processor (202) through an application program interface (API) on web server (116) (Col. 7, lines 20-26). It is noted that the majority of actions taught by Brooke are performed on a server rather than a client. However, it would have been obvious to one of ordinary skill in the art at the time of invention to perform such actions on the client because a client can also host a server and perform all the actions as claimed, on a single machine. A benefit would have been to use such a system for testing prior to deployment.

In regard to dependent Claim 20, Brooke teaches that an XSP script is a document written in XSP and XML that specifies one or more XML documents as input

to XSP processor (202). The output of server pages processing apparatus (200) is an XML object that may be a well-formed markup language document that is ready to be input to a browser, or it may be expressed in another format that is ready for use in a data-interchange process with another system, or for formatting and delivery to an HTML or non-HTML client (Col. 8, lines 7-14; compare with Claim 20, ***"the act of the client formatting the accessed data for viewing comprises the following: an act of the client formatting the accessed data for viewing in Internet Explorer"*** and Claim 21, ***"... an act of the client processing the specific view class to format the data"*** and Claim 22, ***"... an act of the client processing Extensible Markup Language tags in the specific view class"*** and Claim 23, ***"... an act of the view control processing Extensible Style Language tags in the specific view class"***. It is noted that the majority of actions taught by Brooke are performed on a server rather than a client. However, it would have been obvious to one of ordinary skill in the art at the time of invention to perform such actions on the client because a client can also host a server and perform all the actions as claimed, on a single machine. A benefit would have been to use such a system for testing prior to deployment. Brooke also does not specifically teach using Internet Explorer for viewing. However, it would have been obvious to one of ordinary skill in the art at the time of invention to format the data for IE because of its popularity providing the benefit of having properly formatted web pages for the majority of web users.

In regard to dependent Claim 24, Brooke teaches that Fig. 2 shows a block diagram of server pages processing apparatus (200) for generating and processing

XML and XSL documents using XML server pages language (XSP) in accordance with the present invention (Col. 7, lines 20-23; compare with Claim 24, “... ***an act of the client processing Extensible Markup Language tags and Extensible Style Language tags in the specific view class***”). It is noted that the majority of actions taught by Brooke are performed on a server rather than a client. However, it would have been obvious to one of ordinary skill in the art at the time of invention to perform such actions on the client because a client can also host a server and perform all the actions as claimed, on a single machine. A benefit would have been to use such a system for testing prior to deployment.

In regard to dependent Claims 25-28, Brooke teaches that an XSP script is a document written in XSP and XML that specifies one or more XML documents as input to XSP processor (202). The output of server pages processing apparatus (200) is an XML object that may be a well-formed markup language document that is ready to be input to a browser, or it may be expressed in another format that is ready for use in a data-interchange process with another system, or for formatting and delivery to an HTML or non-HTML client (Col. 8, lines 7-14; compare with Claim 25, “... ***an act of the client formatting the accessed data into data that includes Hypertext Markup Language tags***” and Claim 26, “... ***an act of the client outputting the formatted data***” and Claim 27, “... ***an act of the client displaying the data to a web browser***” and Claim 28, “... ***an act of the client outputting the data in Hypertext Markup Language***”). It is noted that the majority of actions taught by Brooke are performed on a server rather than a client. However, it would have been obvious to one of ordinary

skill in the art at the time of invention to perform such actions on the client because a client can also host a server and perform all the actions as claimed, on a single machine. A benefit would have been to use such a system for testing prior to deployment.

In regard to dependent Claim 30 (and similarly dependent Claim 34), Brooke teaches that an XSP script is a document written in XSP and XML that specifies one or more XML documents as input to XSP processor (202). The output of server pages processing apparatus (200) is an XML object that may be a well-formed markup language document that is ready to be input to a browser, or it may be expressed in another format that is ready for use in a data-interchange process with another system, or for formatting and delivery to an HTML or non-HTML client (Col. 7, lines 7-14; compare with Claim 30 (and similarly Claim 34), “... ***an act of converting the identified stored data into displayable data that includes Hypertext Markup Language tags***”). It is noted that the majority of actions taught by Brooke are performed on a server rather than a client. However, it would have been obvious to one of ordinary skill in the art at the time of invention to perform such actions on the client because a client can also host a server and perform all the actions as claimed, on a single machine. A benefit would have been to use such a system for testing prior to deployment.

In regard to dependent Claim 31 (and similarly dependent Claim 35), Brooke teaches XSP scripts, such as XSP script (300), that are used to generate classes of web pages. In one embodiment, XSP script (300) is initiated in XSP processor (202) by



a user or another program requesting the URL of the script, such as the URL `http://server/xsp/inventors.xsp?target=ie5`, where, in this example, the source for the script is stored in a file named `inventors.xsp` in the `xsp` directory on computer system server. The request also includes parameters that provide information pertaining to, among other things, the XML and XSL documents that will be generated during data gathering process (204) and style sheet builder process (206) shown in Fig. 2 (Col. 7, lines 53-64; compare with Claim 31 (and similarly Claim 35), “... ***an act of converting the identified stored data into a format that is displayable on a web browser***”). It is noted that the majority of actions taught by Brooke are performed on a server rather than a client. However, it would have been obvious to one of ordinary skill in the art at the time of invention to perform such actions on the client because a client can also host a server and perform all the actions as claimed, on a single machine. A benefit would have been to use such a system for testing prior to deployment.

In regard to dependent Claim 33, Brooke teaches that typically computer system (130) also includes cache memory (150) to facilitate quicker access between processor (132) and main memory (136). I/O peripheral devices often include speaker systems (152), graphics devices (154), and other I/O devices (144) such as display monitors, keyboards, mouse-type input devices, floppy and hard disk drives, DVD drives, CD-ROM drives, and printers (Col. 4, lines 55-62; compare with Claim 33, “... ***the computer-readable medium is a physical storage device***”). It is noted that the majority of actions taught by Brooke are performed on a server rather than a client. However, it would have been obvious to one of ordinary skill in the art at the time of

Art Unit: 2176

invention to perform such actions on the client because a client can also host a server and perform all the actions as claimed, on a single machine. A benefit would have been to use such a system for testing prior to deployment.


**Conclusion**

Any inquiry concerning this communication or earlier communications from the examiner should be directed to James H Blackwell whose telephone number is 703-305-0940. The examiner can normally be reached on Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Joseph H Feild can be reached on 703-305-9792. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

James H. Blackwell  
07/07/04

  
**SANJIV SHAH**  
PRIMARY EXAMINER